

УДК 681.518.3

Д.М. Семенко, студент гр. ПА-91мп, к.т.н., доц. Стаценко О.В.
КПІ ім. Ігоря Сікорського

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ КОНТРОЛЮ ЯКОСТІ ПОВІТРЯ В СИСТЕМАХ ВЕНТИЛЯЦІЇ ІНТЕЛЕКТУАЛЬНИХ БУДІВЕЛЬ

Анотація. У статті розглянута структура системи вентиляції інтелектуальних будівель та визначені вимоги до керування такими системами. Проаналізовані існуючі підходи до передачі даних в таких системах, та визначені найбільш підходящі інструменти до розробки програмного забезпечення цих систем.

Ключові слова: вентиляція, система контролю, передача та відображення даних.

ВСТУП

На сьогодні існує декілька різних визначень поняття «інтелектуальна будівля» [1]. Згідно з одним з них, інтелектуальною будівлею є така будівля, в якій використовуються спеціальні технології для створення більш безпечного, комфортного та продуктивного середовища для мешканців, та збільшується ефективність її експлуатації для власників. Більшість сучасних громадських та житлових будівель планується з метою зменшення витрат за рахунок зменшення впровадження енергоефективних технологій. Одним з основних способів зменшення витрат енергії є використання вентиляційних систем, керування якими здійснюється в залежності від потреби [2]. До складу вентиляційних систем інтелектуальних будівель входять:

- повітроводи з встановленими вентиляторами, швидкість обертання яких, а, відповідно, і продуктивність, регулюється за допомогою автоматизованих електроприводів;
- комплекти обладнання в окремих приміщеннях, які складаються з керованих заслінок, датчиків вмісту вуглекислого повітря та спеціалізованих контролерів;
- системи моніторингу параметрів роботи всіх інженерних мереж будівлі.

Для керування такою системою вентиляції та контролюванням її роботи необхідним є створення системи збору даних про параметри якості повітря в окремих приміщеннях будівлі та параметри стану кожного з елементів системи вентиляції. Вирішення такої задачі передбачає визначення структури апаратної складової частини системи та розробку програмної складової. Апаратна частина може бути побудована на базі спеціалізованих контролерів, встановлених в окремих приміщеннях та пристроїв керування електроприводами, об'єднаними за допомогою локальної мережі. Для розробки програмної складової необхідним є вибір механізму обміну даними, а також визначення найбільш ефективного інструментарію для безпосередньої розробки програмного додатку.

ПОСТАНОВКА ЗАДАЧІ

Вибір та обґрунтування способу передачі даних та розробка програмного клієнтського додатку для відображення параметрів роботи автоматизованої системи керування вентиляцією інтелектуальних будівель.

ОСНОВНИЙ МАТЕРІАЛ

Обмін даними з використанням локальних або глобальних мереж наразі здійснюється з використанням на транспортному рівні протоколу TCP (Transmission Control Protocol – протокол керування передачею), а на прикладному рівні протоколу HTTP (HyperText Transfer Protocol – протокол передачі гіпертексту). Протокол HTTP був створений для зв'язку між веб-браузерами і веб-серверами, але в принципі він може використовуватися і для інших цілей [3].

Широке використання даного протоколу призвело до формування різних підходів його використання, найбільш розповсюдженими серед яких є: HTTP Polling та HTTP Long Polling.

HTTP Polling є дещо застарілим підходом, але він все ж таки ще використовується. Суть передачі даних з використанням такого підходу полягає у послідовному формуванні запитів даних через деякий проміжок часу. У цього методу є багато мінусів. Самі явні з них:

- на частину запитів приходять «порожні» відповіді, що призводить до непродуктивного використання мережі;
- часова затримка між відправленнями запитів призводить до затримки отримання відповідей, що може негативно вплинути на роботу системи.

Покращений варіант попереднього методу HTTP Long Polling полягає в тому, що клієнт відправляє запит на сервер, сервер тримає відкритим з'єднання поки не прийдуть якісь дані або клієнт не відключиться самостійно. Як тільки дані прийшли – відправляється відповідь і з'єднання закривається. Переваги даного методу порівняно з HTTP Polling:

- зменшується кількість запитів, оскільки відсутні «порожні» відповіді;
- підвищується часова точність подій, оскільки передача даних ініціюється клієнтом;
- сервер зберігає події тільки на час перепідключення.

Але для організації обміну даними в розглядуваному випадку існує краще рішення, яке полягає у використанні протоколу прикладного рівня WebSocket. Це технологія, що дозволяє відкрити постійне двонаправлене мережеве з'єднання між браузером користувача та сервером. За допомогою його API (Application Programming Interface) можна відправити повідомлення на сервер і отримати відповідь без виконання http-запиту, причому цей процес буде подієво-керованим [4]. Переваги в порівнянні з HTTP Long Polling:

- встановлюється лише одне з'єднання;
- забезпечується гранично висока часова точність подій.

При виборі такого протоколу розробку серверного програмного забезпечення доцільно здійснювати з використанням платформи .NET та бібліотеки SignalR. Ця бібліотека надає простий API для створення віддалених викликів процедур від сервера до клієнта, які викликають функції JavaScript у браузерах клієнтів (та інших клієнтських платформах) із коду .NET на стороні сервера. SignalR також включає API для управління з'єднаннями та групування

з'єднань. SignalR використовує протокол WebSocket там, де він доступний, і повертається до старих протоколів, де це необхідно [5].

Для розробки клієнтського програмного забезпечення разом із обраною бібліотекою доцільним є використання фреймворку Blazor. Він представляє UI-фреймворк для створення інтерактивних додатків, які можуть працювати як на стороні сервера, так і на стороні клієнта, на платформі .NET.

Blazor надає розробникам наступні переваги:

- написання коду веб-додатків за допомогою C # замість JavaScript;
- використання можливостей екосистеми .NET, зокрема, бібліотек .NET при створенні додатків;
- використання Visual Studio в якості інструменту для розробки, який має вбудовані шаблони для спрощення створення додатків.

З урахуванням проведеного аналізу структура системи має такий вигляд:

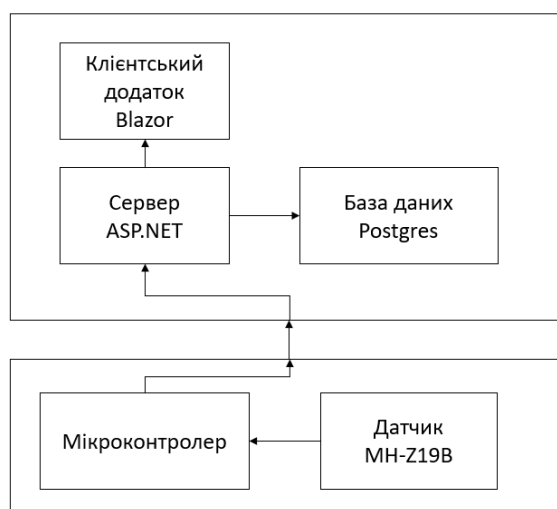


Рисунок 1 – Структурна схема системи контролю якості повітря

На рис.1 структурно показаний зв'язок програмної складової з вимірювальним модулем в одному приміщенні будівлі, при цьому передбачається, що вимірювальні модулі в інших приміщеннях мають таку саму структуру, а під час передачі даних використовують інші ідентифікатори.

Програмний продукт складається з таких компонентів: бази даних Postgres, серверної частини ASP.NET та клієнтського додатку. Дані від датчиків та параметри роботи системи вентиляції передаються з використанням мікроконтролерів до серверу. Реляційна база даних використовується для зберігання цих даних, а також списку датчиків. Сервер оброблює ці дані та вираховує потужність вентилятора, яку треба встановити в тому чи іншому приміщенні будівлі.

Розроблений клієнтський додаток має декілька складових частин, які мають вигляд наведений на рис.2,3.

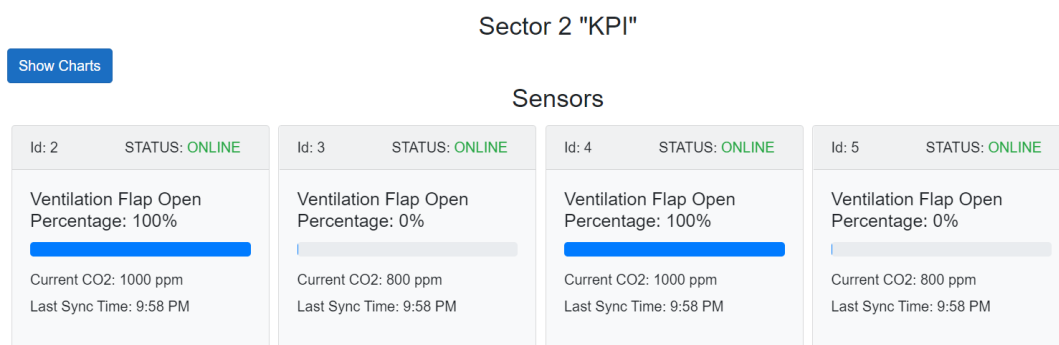


Рисунок 2 – Список датчиків в приміщеннях одного сектору з виводом поточних даних

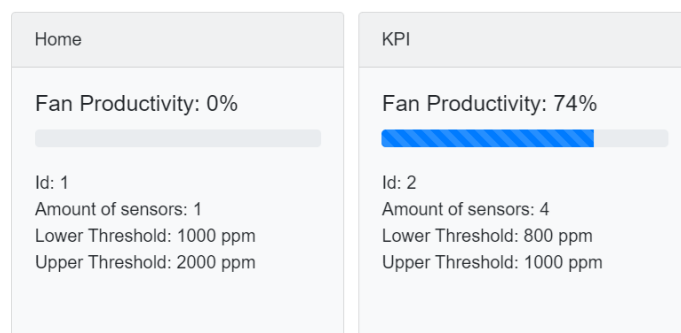


Рисунок 3 – Список секторів та відображення параметрів роботи системи вентиляції

ВИСНОВОК

В роботі проведений аналіз протоколом передачі даних, що можуть бути використані при побудові системи керування та діагностики роботи вентиляції інтелектуальних будівель, а також розроблений клієнтський програмний додаток на основі фреймворку Blazor.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- [1] Osama Omar. Intelligent building, definitions, factors and evaluation criteria of selection // Alexandria Engineering Journal, (2018) 57, pp.2903–2910.
- [2] Семенко Д.М. Енергозбереження засобами автоматизованих вентиляційних систем / Д.М. Семенко, О.В. Стаценко // Збірник праць XIII Всеукраїнської науково-практичної конференції студентів, аспірантів та молодих вчених “ПОГЛЯД У МАЙБУТНЄ ПРИЛАДОБУДУВАННЯ” – К.: ПБФ, КПІ ім. Ігоря Сікорського, Центр учбової літератури. – 2020. – С.389-392.
- [3] Приложения реального времени и Polling ,Long Polling , WebSockets, Server-Sent Events SSE.// Режим доступа: <https://intellect.icu/prilozheniya-realnogo-vremeni-i-polling-long-polling-websockets-server-sent-events-sse-7016/>
- [4] WebSockets. MDN Web Docs. // Режим доступа: <https://developer.mozilla.org/ru/docs/WebSockets/>
- [5] Patrick Fletcher. Introduction to SignalR. // Режим доступа: <https://docs.microsoft.com/en-us/aspnet/signalr/overview/getting-started/introduction-to-signalr>

Наук. керівник – к.т.н., доц. Стаценко О.В.